



TUGAS AKHIR – TF 145565

**RANCANG BANGUN SISTEM PENGENDAIAN LEVEL
PADA PLANT SIMULATOR PENGENDALIAN FLOW
DAN LEVEL**

**Zumrotun Naimah
NRP 105 115 00000 063**

**Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001**

**Dosen Pembimbing II
Ahmad Fauzan Adziimaa, ST. MSc
NIP. 1991201711052**

**POGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**



TUGAS AKHIR – TF 145565

**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL
PADA PLANT SIMULATOR PENGENDALIAN FLOW
DAN LEVEL**

**Zumrotun Naimah
NRP 105 115 00000 063**

**Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001**

**Dosen Pembimbing II
Ahmad Fauzan Adziimaa, ST. MSc
NIP. 1991201711052**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**



FINAL PROJECT - TF 145565

***DESIGN CONTROL SYSTEM LEVEL IN FLOW AND
LEVEL SIMULATOR PLANT***

Zumrotun Naimah
NRP 105 115 00000 063

Supervisor I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001

Supervisor II
Ahmad Fauzan Adziimaa, ST. MSc
NIP. 1991201711052

***DIII INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018***

**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL
PADA PLANT SIMULATOR PENGENDALIAN FLOW
DAN LEVEL**

TUGAS AKHIR

Oleh :

**Zumrotun Naimah
NRP 105 115 00000 063**

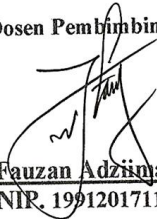
**Surabaya, 25 Juli 2018
Mengetahui dan Menyetujui,**

Dosen Pembimbing I,



Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001

Dosen Pembimbing II,



Ahmad Fauzan Adzima, ST. MSc
NIP. 1991201711052



**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL
PADA PLANT SIMULATOR PENGENDALIAN FLOW
DAN LEVEL**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Ahli Madya
Pada Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

**Zumrotun Naimah
NRP 105 115 00000 063**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Totok Soehartanto, DEA..... (Dosen Pembimbing I)
2. Ahmad Fauzan Adziimaa, ST, MSc..... (Dosen Pembimbing II)
3. Ir. Tutug Dhanardono, MT..... (Dosen Penguji I)

SURABAYA
JULI 2018

RANCANG BANGUN SISTEM PENGENDALIAN LEVEL PADA PLANT SIMULATOR PENGENDALIAN FLOW DAN LEVEL

Nama : Zumrotun Naimah
NRP : 10511500000063
Jurusan : DIII Teknologi Instrumentasi, ITS Surabaya
Pembimbing : Dr. Ir. Totok Soehartanto, DEA
Ahmad Fauzan Adziimaa, ST. MSc

Abstrak

Pada tugas akhir ini dilakukan perancangan sistem pengendalian level pada plant simulator flow dan level. Pada program pendidikan vokasi yang sebagian besar merupakan praktikum, maka dari itu dibutuhkan sarana praktikum yang mendukung mata kuliah tertentu seperti pada mata kuliah Teknik Otomasi. Pada sistem pengendalian level ini, terdiri dari tiga elemen penting yaitu kontroler, *final control element*, dan sensor. Sensor yang digunakan untuk mengukur level adalah sensor *Ultrasonic*, kontroler yang digunakan pada sistem pengendalian ini yaitu mikrokontroler atmega 16, dan aktuator (*final control element*) berupa motor valve. Pengendalian level ini termasuk *feedback close loop control*, dimana ketika *output* yang dihasilkan tidak sesuai dengan *set point* yang telah ditentukan maka sensor akan mengirimkan sinyal kepada kontroler berupa mikrokontroler atmega 16 yang kemudian akan diproses dan memberikan perintah kepada aktuator agar *output* yang dihasilkan sesuai dengan *set point*. Respon pengendalian yang didapat menunjukkan bahwa pengendalian telah berjalan sesuai dengan *setpoint* yang diinginkan yaitu 20 cm dan beresilasi mendekati *setpoint*. Selain itu berdasarkan pengujian alat pada sistem pengendalian level didapatkan *respon time* yang cukup baik untuk mencapai *steady state*. Waktu yang dibutuhkan untuk mencapai *set point* (*settling time*) dari awal pengambilan data adalah ± 34 detik.

Kata kunci : *Simulator, Level Control, Flow Control.*

DESIGN CONTROL SYSTEM LEVEL IN FLOW AND LEVEL SIMULATOR PLANT

Name : Zumrotun Naimah
NRP : 10511500000063
Departement : *Diploma of Instrumentation Engineering, ITS
Surabaya*
Supervisor : **Dr. Ir. Totok Soehartanto, DEA**
Ahmad Fauzan Adziima, ST. MSc

Abstract

On this final assignment done designing control systems level on plant simulators flow and level. On the education program students are largely a practical, therefore it takes a practical means of supporting specific courses such as in Automation Engineering courses. At this level, the control system consists of three key elements, namely controller, final control element, and sensors. The sensors used to measure the level of Ultrasonic sensors, the controller is used in these control systems namely atmega microcontroller and actuators (final control element) in the form of motor valve. This level includes the control feedback loop close control, which when the output produced does not match the specified set point then the sensor will send a signal to the controller atmega microcontroller form 16 which will then processed and give commands to the actuator so that the output is generated in accordance with the set point. The response of the control indicates that the control has been running in accordance with the desired setpoint i.e. 20 cm and oscillating approaching the setpoint. Besides testing tool based on control systems level obtained response time is good enough to reach steady state. The time it takes to reach the set point (settling time) from the initial data retrieval is ± 34 seconds.

Key Words: *Simulator, Level Control, Flow Control*

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat, rahmat dan kebesaran-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “RANCANG BANGUN SISTEM PENGEDALIAN LEVEL PADA PLANT SIMULATOR PENGENDALIAN FLOW DAN LEVEL” tepat pada waktunya.

Selama menyelesaikan tugas akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Orang tua tercinta yang telah memberikan segala dukungan baik moral maupun materil serta dukungan yang sangat luar biasa
2. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Kepala Departmen Teknik Instrumentasi ITS yang telah memberikan semangat dan motivasi kepada kami
3. Bapak Detak Yan Pratama selaku dosen wali kami
4. Bapak Dr. Ir. Totok Soehartanto, DEA dan Bapak Ahmad Fauzan Adziimaa, ST. MSc selaku dosen pembimbing yang telah meluangkan waktu, memberikan arahan dan saran yang membantu selama proses pengerjaan tugas akhir ini
5. Terimakasih kepada Mas Nafi' yang menjadi bagian dari anggota Simulator Flow dan Level yang bersama-sama berjuang dalam pengerjaan tugas akhir ini
6. Teman-teman tercinta Teknik Intrumentasi 2015 yang senantiasa memberikan semangat,
7. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa kesempurnaan hanya milik Allah SWT. Oleh sebab itu, penulis sangat berterimakasih atas segala masukan, kritik dan saran yang membangun dari pembaca agar laporan ini menjadi lebih baik untuk di kemudian hari. Demikian

laporan ini penulis buat, semoga laporan ini dapat memberikan manfaat selain bagi penulis sendiri, dan bagi pembaca sekalian.

Surabaya, 25 Juli 2018

Penulis

Zumrotun Naimah
NRP. 10511500000063

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
ABSTRAK.....	v
ABSTRACT.....	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii

BAB I. PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Manfaat.....	2
1.6 Sistematika Laporan.....	3

BAB II. TINJAUAN PUSTAKA

2.1 Sistem Pengendalian Level.....	5
2.2 Pengendali Proporsional.....	6
2.2 Sensor Ultrasonic.....	7
2.3 <i>Water Flow Sensor</i>	8
2.4 <i>Stepper Motor Controlled Valve</i>	10
2.5 Mikrokontroler Atmega16.....	11
2.6 Code Vision AVR.....	14

BAB III. PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir (<i>Flowchart</i>)	15
3.2 Perancangan Sistem.....	16

BAB IV. HASIL DATA DAN PEMBAHASAN

4.1 Rancang Bangun Alat	23
4.2 Pengujian Sistem	30
4.3 Pembahasan	34

BAB V. PENUTUP

5.1 Kesimpulan.....	36
5.2 Saran.....	36

DAFTAR PUSTAKA

BIODATA PENULIS

DAFTAR GAMBAR

	Hal
Gambar 2.1 Diagram Blok Sistem Pengendalian Level	5
Gambar 2.2 Skema Cara Kerja Sensor Ultrasonic	8
Gambar 2.3 <i>Flowmeter Turbin</i>	8
Gambar 2.4 Skematik instalasi water flow sensor	9
Gambar 2.5 Mechanic Dimensi <i>Water Flow Sensor</i> G1/2.....	9
Gambar 2.6 Motor stepper dengan lilitan unipolar	10
Gambar 2.7 <i>Mikrokontroler ATmega16</i>	12
Gambar 2.8 Pin-pin AVR ATmega 16	12
Gambar 3.1 Diagram alir Pembuatan Alat.....	15
Gambar 3.2 <i>Process Flow Diagram</i>	16
Gambar 3.3 <i>Piping and Instrument Diagram</i>	17
Gambar 3.4 Diagram Blok Pengendalian Level.....	18
Gambar 3.5 <i>Design Plant</i>	18
Gambar 3.6 Desain Tangki Pengukuran.....	18
Gambar 3.7 Desain Tangki Pengukuran	19
Gambar 3.8 Pompa Aquarium	20
Gambar 3.9 <i>Motor Operating Valve</i>	20
Gambar 3.10 <i>Water Flow Sensor</i>	21
Gambar 3.11 Sensor Ultrasonic	21
Gambar 3.12 Tampilan <i>Software Code Vision AVR V 2.05.0</i>	22
Gambar 4.1 Rancang Bangun Plant.....	23
Gambar 4.2 Realisasi Pemasangan Pengendalian Level.....	24
Gambar 4.3 Grafik Pembacaan oleh Sensor dan Penggaris.....	27
Gambar 4.4 Grafik Pengujian Sensor Flow	30
Gambar 4.5 Grafik Respon Dinamik.....	32
Gambar 4.6 Grafik Respon Valve.....	33

DAFTAR TABEL

	Hal
Tabel 2.1 Pergerakan Full Step.....	11
Tabel 2.2 Fungsi khusus port B.....	13
Tabel 2.3 Fungsi khusus port C.....	13
Tabel 2.4 Fungsi khusus port D.....	14
Tabel 4.1 Hasil Pengujian Stepper	24
Tabel 4.2 Pengambilan Data Naik.....	25
Tabel 4.3 Pengambilan Data Turun.....	26
Tabel 4.4 Data Kalibrasi Sensor.....	26
Tabel 4.5 Data Kalibrasi Lanjutan	26
Tabel 4.6 Data Pengukuran pada Tangki Ukur	28
Tabel 4.7 Hasil Pengukuran Oleh Sensor dan Perhitungan.....	28
Tabel 4.8 Hasil Pengujian Sistem Pengendalian Level.....	29
Tabel 4.9 Respon <i>Valve</i> Terhadap Nilai Level.....	32

BAB I

PENDAHULUAN

1.1. Latar Belakang

Metode pembelajaran yang dipusatkan pada penerapan keilmuan berdasarkan praktek sangat dibutuhkan oleh sebuah program pendidikan vokasi. Maka dari itu dibutuhkan sarana praktikum yang mendukung mata kuliah tertentu seperti pada mata kuliah Teknik Otomasi, guna untuk mendukung pengajaran dan untuk menunjang kompetensi. Simulator yang saat ini ada di pasaran masih terbilang mahal, selain mahal simulator yang ada di pasaran juga konvensional dan sebagian masih menggunakan prinsip melalui penggunaan peralatan kontrol mekanis, pegas, atau prinsip instrumentasi dasar yang masih sederhana. Sistem otomatis dan adanya *monitoring variable* di dalam proses plant tersebut sangat diperlukan agar bisa berjalan lebih efektif dan efisien.

Sehingga ide untuk menjawab permasalahan diatas maka dalam tugas akhir ini akan dibuat simulator plant untuk menunjang praktikum dari mata kuliah tertentu yaitu perancangan dan pembuatan sistem pengendalian level pada plant simulator flow dan level. Plant simulator ini dibuat dengan spesifikasi yang sesuai, akan tetapi dengan harga yang lebih terjangkau. Selain itu, plant simulator ini akan dibuat lebih *full automatic* dengan banyak sensor dan instrumentasi di dalamnya, ditambah dengan adanya monitoring dalam plant tersebut. Dengan adanya sebuah *transmitter* yang berguna untuk mengukur nilai proses lalu mengirimkannya ke instrument lain yaitu *controller* dimana *controller* berfungsi untuk membandingkan nilai proses terhadap nilai yang diinginkan dan kemudian akan menghitung *manipulated variable* yang diperlukan untuk mengatur *final control element*. Sehingga dengan adanya media transmisi maka elemen-elemen tersebut dapat berkomunikasi antar elemen satu dan elemen yang lain secara otomatis.

Dengan adanya Tugas Akhir ini maka akan dihasilkan sebuah plant simulator untuk menunjang sarana praktikum dari satu atau lebih mata kuliah. Hasil daripada tugas akhir ini akan dapat

menambah pemahaman siswa terhadap konsep *feedback close loop control*. Serta adanya peluang bagi mahasiswa untuk turut serta dalam mengembangkan laboratorium Departemen Teknik Instrumentasi.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana merancang dan membuat sistem pengendalian level menggunakan *feedback close loop control*?
2. Bagaimana mengetahui respon dari simulator pengendalian level menggunakan *feedback close loop control*?

1.3. Tujuan

Berdasarkan rumusan permasalahan yang dijelaskan diatas, maka tujuan dari Tugas Akhir ini adalah :

1. Untuk merancang dan membangun sistem pengendalian level menggunakan *feedback close loop control*.
2. Untuk mengetahui respon dari pengendalian level menggunakan *feedback close loop control*.

1.4. Lingkup Tugas Akhir

Untuk memfokuskan penyelesaian masalah pada penelitian tugas akhir ini maka lingkup tugas akhir yang diangkat adalah sebagai berikut:

1. Sistem pengendalian level ini menggunakan *feedback close loop control*.
2. Indikator atau monitoring langsung terhubung menjadi satu di sebuah *display*.

1.5. Manfaat

Manfaat dari tugas akhir ini adalah sebagai berikut :

1. Sebagai penunjang praktikum bagi mahasiswa Teknik Instrumentasi pada beberapa mata kuliah seperti Teknik Otomasi, Sistem Pengukuran Aliran, dan Sistem Instrumentasi Industri.

1.6 Sistematika Laporan

Dalam Penyusunan tugas akhir ini, sistematika laporan akan disusun secara sistematis yang terbagi dalam beberapa bab, yakni dengan perincian sebagai berikut :

BAB I PENDAHULUAN

Bab I ini terdiri dari penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika laporan

BAB II Tinjauan Pustaka

Pada bab II ini berisikan tentang teori-teori penunjang tugas akhir, antara lain teori tentang sistem pengendalian proses, *water flow sensor*, *Motor Stepper*, Mikrokontroller Atmega, Ultrasonic, Solenoid Valve.

BAB III Perancangan dan Pembuatan Alat

Dalam bab ini terdiri dari penjelasan secara detail langkah-langkah yang dilakukan untuk mencapai tujuan dari penelitian. Produk yang dihasilkan dari tahap-tahap yang telah dilakukan adalah perancangan dan model yang siap untuk dibuat, diuji dan dianalisa

BAB IV Pengujian Alat dan Analisa Data

Pada bab ini merupakan tindak lanjut dari bab III, dimana jika telah melaksanakan perancangan dan pembuatan alat maka dilakukan pengujian alat sehingga memperoleh data baik data berupa grafik maupun tabulasi, kemudian dilakukan analisa data dan pembahasan

BAB V Kesimpulan dan saran

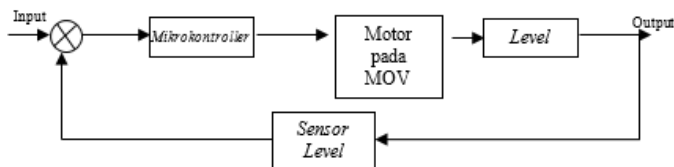
Dalam bab ini adalah berisi mengenai kesimpulan pokok dari keseluruhan rangkaian penelitian yang telah dilakukan serta saran yang dapat dijadikan rekomendasi sebagai pengembangan penelitian selanjutnya.

Halaman ini sengaja dikosongkan

BAB II DASAR TEORI

2.1 Sistem Pengendalian Level

Sistem kontrol atau sistem pengendalian adalah proses pengaturan / pengendalian terhadap satu atau beberapa besaran (variabel, parameter) sehingga berada pada suatu harga atau dalam suatu rangkuman harga (*range*) tertentu. Dalam istilah lain disebut juga teknik pengaturan, sistem pengendalian atau sistem pengontrolan. Salah satu penerapan sistem pengendalian adalah untuk menjaga ketinggian fluida pada suatu wadah. Berikut adalah diagram blok sistem pengendalian level.



Gambar 2.1 Diagram Blok Sistem Pengendalian Level^[1].

Sensor adalah elemen sistem yang secara efektif berhubungan dengan proses dimana suatu variabel sedang diukur dan menghasilkan suatu keluaran dalam bentuk tertentu tergantung pada variabel masukannya, dan dapat digunakan oleh bagian sistem pengukuran yang lain untuk mengenali nilai variabel tersebut.

Kontroler merupakan elemen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (set point) atau sinyal error mejadi sinyal kontrol. Sinyal error disini adalah selisih antara sinyal umpan balik yang dapat berupa sinyal keluaran plant sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (setpoint).

Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem.

Aktuator diaktifkan dengan menggunakan lengan mekanis yang biasanya digerakkan oleh motor listrik, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroler. Aktuator adalah elemen yang mengkonversikan besaran listrik analog menjadi besaran lainnya misalnya kecepatan putaran dan merupakan perangkat elektromagnetik yang menghasilkan daya gerakan sehingga dapat menghasilkan gerakan pada robot. Untuk meningkatkan tenaga mekanik aktuator ini dapat dipasang sistem gearbox. Aktuator dapat melakukan hal tertentu setelah mendapat perintah dari kontroller.

Plant adalah nama lain untuk sistem. Plant adalah seperangkat peralatan yang hanya terdiri dari beberapa bagian dan bekerja bersama-sama, yang digunakan untuk melakukan suatu operasi tertentu. Pada sistem pengendalian, setiap obyek fisik yang dikontrol disebut plant^[1].

2.2 Pengendali *Proporsional*

Proporsional adalah persen perubahan sinyal kendali sebanding dengan persen perubahan sinyal pengukuran. Dengan kata lain sinyal kendali merupakan kelipatan sinyal pengukuran. Respon proporsional merupakan dasar pengendali PID. Pemakaian pengendali proporsional selalu menghasilkan offset. Offset berarti pengendali mempertahankan nilai PV pada suatu harga yang berbeda dengan setpoint. Offset muncul dalam usaha pengendali mempertahankan keseimbangan massa dan/atau energi. Pengendali proporsional hanya dapat digunakan untuk proses yang dapat menerima offset. Faktor kelipatan disebut gain pengendali (Kc). Pengendali proporsional sebanding dengan error-nya^[2].

Persamaan matematika :

$$U = K_c \cdot e + U_o$$

dengan,

U = Keluaran pengendali (sinyal kendali),

Kc = Proportional gain (gain pengendali)

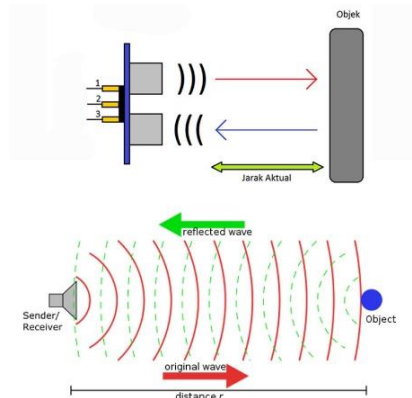
e = Error (SP – PV)

Uo = bisa, yaitu nilai sinyal kendali saat tidak ada error (e = 0). Istilah gain pengendali bisaanya dinyatakan

dalam proportional band (PB) Harga PB berkisar 0 – 500. PB pada dasarnya menunjukkan persentasi rentang PV yang dapat dikendalikan atau *range error* maksimum sebagai masukan pengendali yang dapat menyebabkan pengendali memberikan keluaran dengan range maksimum. Semakin sempit proportional band, offset semakin kecil yang sesuai dengan proses dengan kapasitas besar, waktu mati kecil sehingga dapat memakai proportional band yang sempit^[2].

2.3 Sensor Ultrasonic

Sensor *ultrasonic* adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik). Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa. Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima^[3].



Gambar 2.2 Skema Cara Kerja Sensor Ultrasonic^[3]

2.5 Water Flow Sensor

Flowmeters turbin menggunakan energi mekanik dari cairan yang mengalir guna memutar sebuah “*pinwheel*” (rotor). *Blades* pada rotor dibuat miring dengan sudut tertentu layaknya baling-baling, untuk mengubah energi dari aliran fluid menjadi energi rotasi. Poros rotor berputar pada *bushing/bearing* dimana ketika liquid bergerak lebih cepat pada *flowmeter turbin*, maka akan lebih banyak menghasilkan pulsa yang akan sebanding dengan kecepatan aliran fluida. Sensor *pick-up* pada *flowmeter turbin* akan memproses sinyal pulsa untuk menentukan aliran fluida.^[4]



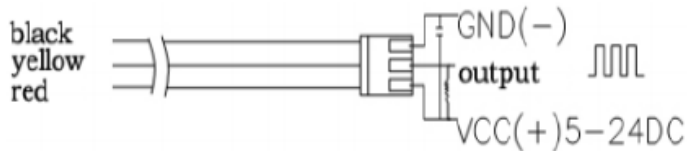
Gambar 2.3 *Flowmeter Turbin*^[4]

Dimana output dari *water flowmeter* ini yaitu frekuensi.
 Persamaan untuk merubah frekuensi ke debit yaitu :

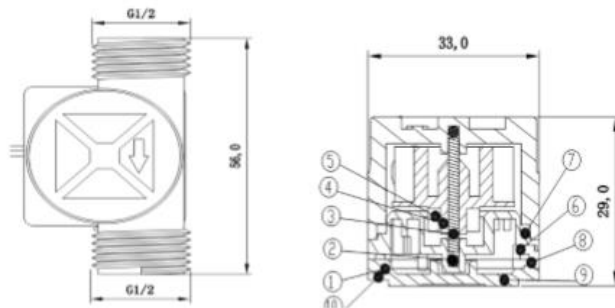
$$F(\text{Hz}) = 4.8 \times Q (\text{L/min})$$

Spesifikasi G 1/2 inch *water flow sensor* :

- Rentang tegangan operasional antara 5 ~ 24 Volt DC
- Maksimum arus operasional sebesar 15 mA (pada 5 VDC)
- Maksimum suhu air 120°C
- Maksimum suhu operasional 80°C
- Rentang kelembapan operasional antara 35% ~ 90% RH
- Maksimum tekanan air 1,75 MPa
- Diameter *thread inlet/outlet* 1/2"



Gambar 2.4 Skematik instalasi water flow sensor^[4].

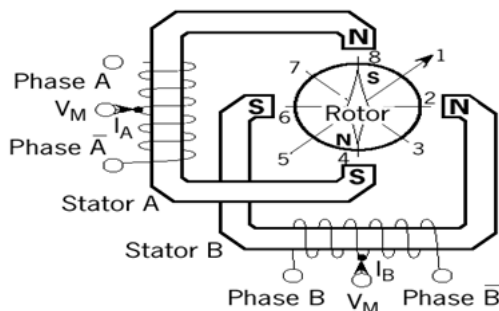


Gambar 2.5 Mechanic Dimensi *Water Flow Sensor* G1/2^[4]

2.6 Stepper Motor Controlled Valve

Motor stepper disini berfungsi sebagai penggerak valve secara elektrik. Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik^[5].


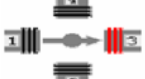
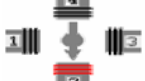
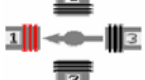
Berdasarkan metode perancangan rangkain pengendalnya, motor stepper dapat dibagi menjadi jenis unipolar dan bipolar. Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (V_M) pada bagian tengah (center tap) dari lilitan^[5].



Gambar 2.6 Motor stepper dengan lilitan unipolar^[5]

Full step adalah cara mengendalikan motor stepper sehingga dihasilkan pergerakan motor namun tidak sehalus pergerakan half step. Hal ini disebabkan karena pergerakan rotor dalam motor stepper yang bergerak per 1 buah kutub (coil). Untuk lebih jelas, konfigurasi motor stepper dengan pengendalian *full step* untuk setiap pergerakan motor adalah sebagai berikut :

Tabel 2.1 Pergerakan Full Step^[5]

Step	Coil 4	Coil 3	Coil 2	Coil 1	Pergerakan Motor
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	

2.7 Mikrokontroler ATmega16

Mikrokontroler merupakan keseluruhan sistem komputer yang dikemas menjadi sebuah *chip* di mana di dalamnya sudah terdapat Mikroprosesor, I/O, Memori bahkan ADC, berbeda dengan Mikroprosesor yang berfungsi sebagai pemroses data.

Mikrokontroller AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock* atau dikenal dengan teknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR dapat dikelompokkan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing adalah kapasitas memori, *peripheral* dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama. ATmega16 memiliki kapasitas EEPROM 1024 bytes dan kapasitas FLASH 16K words (32K bytes) sehingga program yang dibuat dapat memiliki ukuran besar. Gambar 2.28 menunjukkan mikrokontroler AVR ATmega 16 Berikut ini gambar Mikrokontroler Atmega16^[6].

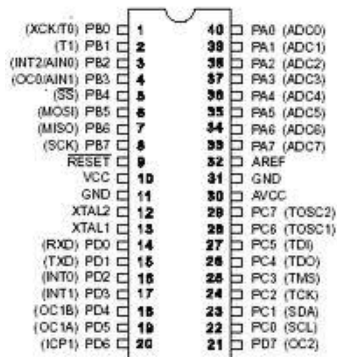


Gambar 2.7 Mikrokontroler ATmega16^[6]

2.7.1 Konfigurasi Pin Atmega16

Konfigurasi pin pada ATmega16 dengan kemasan 40-pin DIP (Dual Inline Package). Kemasan pin tersebut terdiri dari 4 Port yaitu Port A, Port B, Port C dan Port D. Masing masing port terdiri dari 8 buah pin. Selain itu juga terdapat pin RESET, VCC, GND 2 buah, VCC, XTAL1, XTAL2 dan AREF.

Gambar 2.29 menunjukkan pin-pin pada mikrokontroler AVR ATmega 16.



Gambar 2.8 Pin-pin AVR ATmega 16^[6].

Deskripsi dari pin-pin ATmega16 dijelaskan sebagai berikut:

VCC merupakan pin yang berfungsi sebagai masukan catu daya

GND merupakan pin *ground*

PORT A (PA0-PA7) merupakan pin *input/output* dua arah dan pin masukan ADC.

PORT B (PB0-PB7) merupakan pin *input/output* dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.2.

Tabel 2.2 Fungsi khusus port B^[6].

PIN	Fungsi Khusus
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 T1 (Timer/Counter0 External Counter input) XCK (USART External Clock Input/Output)

PORT C (PC0-PC7) merupakan pin *input/output* dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Fungsi khusus port C^[6].

PIN	Fungsi Khusus
PC7	TOSC2 (Timer Oscillator pin2)
PC6	TOSC1 (Timer Oscillator pin1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two – Wire Serial Bus Data Input Output Line)
PC0	SCL (Two – Wire Serial Bus Clock Line)

PORT D (PD0-PD7) merupakan pin input/output dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.4.

Tabel 2.4 Fungsi khusus port D^[6].

PIN	Fungsi Khusus
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

2.8 CodeVision AVR

CodeVision AVR merupakan sebuah *software* yang digunakan untuk memprogram mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah disediakan oleh program tersebut. Salah satu *compiler* program yang umum digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C.

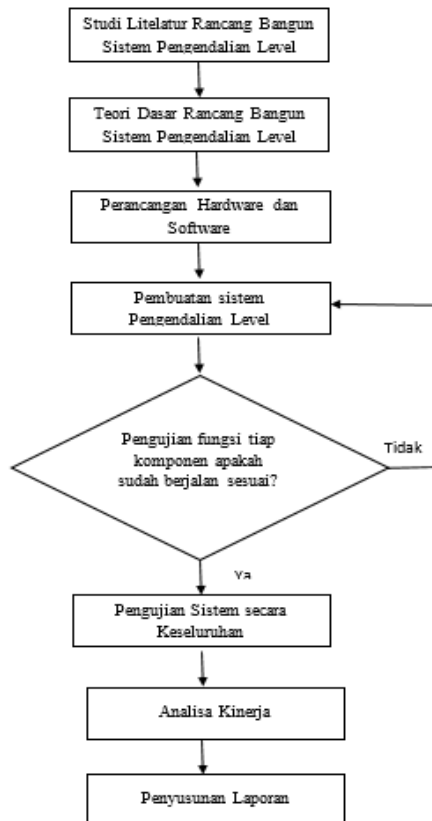
CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan pengguna atau pemrogram dalam inisialisasi mikrokontroler yang akan digunakan^[7].

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Diagram Alir (*Flowchart*)

Pada tugas akhir kali ini membuat sistem pengendalian level pada simulator pengendalian flow dan level. Langkah-langkah perancangan alat ini digambarkan dalam diagram alir penelitian yang dapat dilihat pada gambar berikut ini :

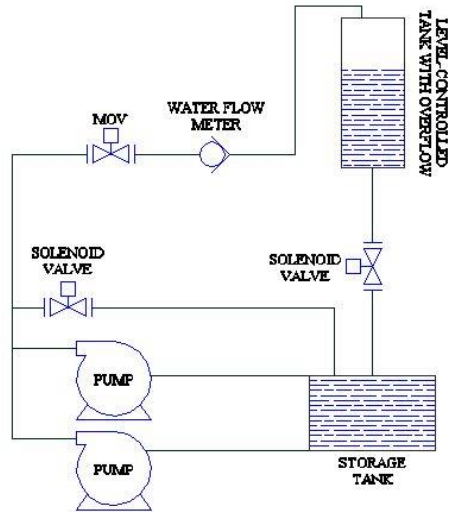


Gambar 3.1 Diagram alir Pembuatan Alat

3.2 Perancangan Sistem

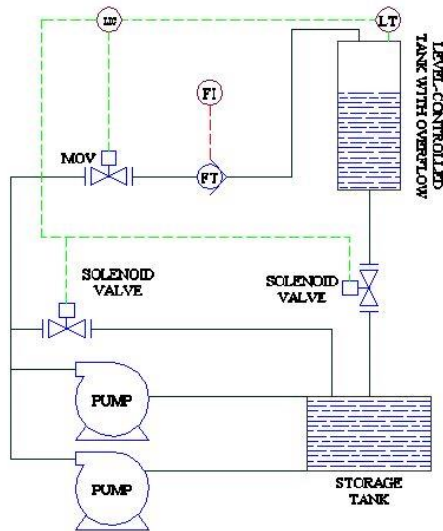
Dilakukan perancangan *hardware* dan *software* pada plant simulator pengendalian level.

3.2.1 Perancangan PFD dan P&ID



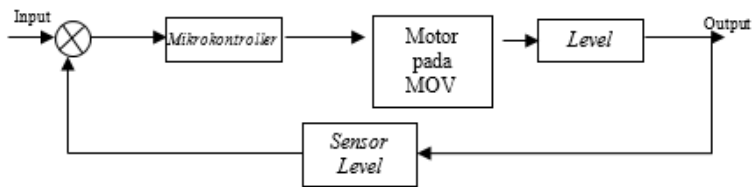
Gambar 3.2 *Process Flow Diagram*

Setelah desain dari *Proses Flow Diagram* dibuat maka langkah selanjutnya yaitu membuat desain *Piping and Instrument Diagram* seperti pada gambar 3.3

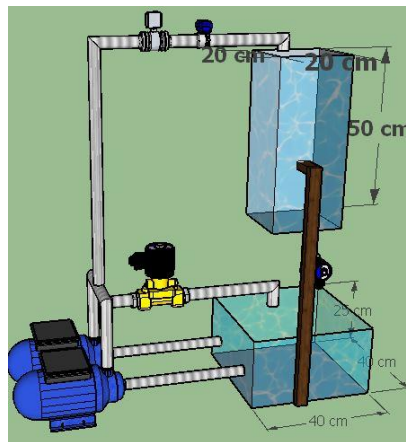


Gambar 3.3 *Piping and Instrument Diagram*

Set point yang telah ditentukan, akan dikirimkan ke mikrokontroller berupa atmega 16 yang kemudian mikrokontroller akan mengirimkan sinyal kepada aktuatur berupa motor valve untuk bekerja sebagaimana mestinya, kemudian sinyal akan di baca oleh sensor atau transmitter untuk kemudian di proses kembali (*feedback*). Apabila telah mencapai *set point*, maka motor valve akan bergerak menutup dan sebaliknya apabila belum mencapai *set point* maka motor valve akan bergerak membuka sampai 100%.

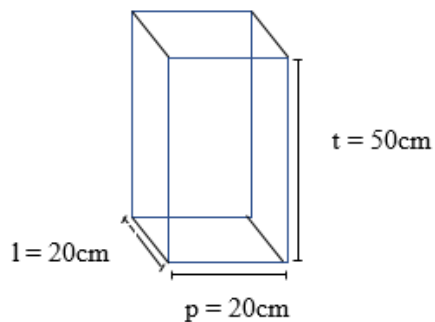


Gambar 3.4 Diagram Blok Sistem Pengendalian Level



Gambar 3.5 Design Plant

3.2.2 Perancangan Hardware



Gambar 3.6 Desain Tangki Pengukuran

Tangki pengukuran yang digunakan adalah bentuk balok dan berbahan dasar akrilik dengan ketebalan 5mm yang mempunyai dimensi sebagai berikut :

$$\begin{aligned}
 Q_{in1} + Q_{in2} &= Q_{out} + \text{akumulasi volume} \\
 Q_{in \text{ total}} &= Q_{out} + \frac{\text{akumulasi volume}}{t} \\
 Q_{in \text{ total}} - Q_{out} &= A \times \frac{dh}{dt} \\
 \frac{Q_{in \text{ total}} - Q_{out}}{A} &= \frac{dh}{dt}
 \end{aligned}$$

Sehingga ukuran tangki sebagai berikut:

$$V \text{ balok} = p \times l \times t$$

$$20.000 \text{ cm}^3 = p \times l \times 50\text{cm}$$

$$400 = p \times l$$

Sehingga :

$$p = 20\text{cm}, l = 20\text{cm}$$

3.2.3 Pembuatan Sistem

- **Pembuatan Hardware**

1. Tangki Pengukuran



Gambar 3. 7 Tangki Pengukuran

Pada sistem ini terdapat 1buah tangki yang terbuat dari bahan *acrylic* dan memiliki kapasitas penampungan 20 L, dengan panjang 20 cm, lebar 20 cm, dan tinggi 50 cm.

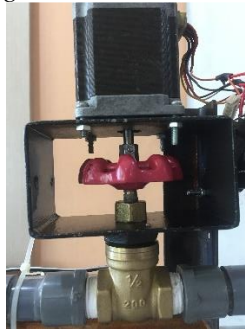
2. Pompa Air



Gambar 3. 8 Pompa Aquarium

Pompa air yang digunakan pada sistem pengendalian leve ini merupakan pompa aquariu yang memiliki spesifikasi 2800 liter per menit.

3. *Motor Operating Valve*



Gambar 3.9 *Motor Operating Valve*

Pada sistem ini terdapat 1 buah Motor Valve yang terletak diantara pompa dan flow meter .Motor Valve ini berfungsi untuk mengatur bukaan/katup yang dialiri air. Motor valve yang digunakan adalah jenis stepper dengan spesifikasi 3A dan 1,8 Degre.

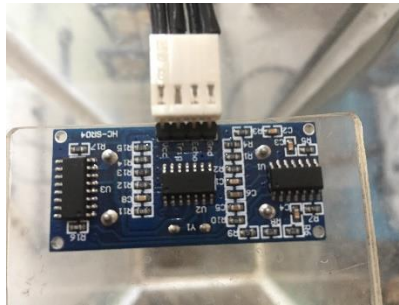
4. *Water Flow Sensor*



Gambar 3.10 *Water Flow Sensor*

Sensor pengukuran flow menggunakan *water flow sensor* G1/2. Rentang tegangan operasional antara 5 ~ 24 Volt DC. Maksimum arus operasional sebesar 15 mA (pada 5 VDC)

5. *Sensor Ultrasonic*



Gambar 3.11 *Sensor Ultrasonic*

Sensor untuk pengukuran level menggunakan sensor ultrasonic HCSR04 dengan spesifikasi *working voltage* 5

3.2.3 **Pembuatan Perangkat Lunak (*Software*)**

Pada perancangan alat ini terdapat *software* yang digunakan untuk membuat program sebelum di-*compile* ke mikrokontroler dan diguqnakan sebagai *compiler* program ke, yaitu *Code Vision AVR V 2.05.0*.

BAB IV

ANALISA DATA DAN PEMBAHASAN

4.1 Rancang Bangun Alat

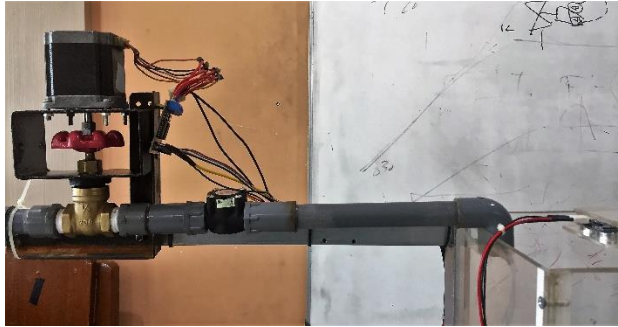
Berikut ini adalah rancang bangun Plant Simulator Pengendalian Level



Gambar 4. 1 Rancang Bangun Plant Simulator Pengendalian Level

4.1.1 Pembuatan Sistem Pengendalian Level

Berikut ini adalah perancangan sistem pengendalian level pada tangki penampungan plant simulator pengendalian flow dan level.



Gambar 4. 2 Realisasi Pemasangan sistem Pengendalian Level

4.1.2 Pengujian Komponen

Pengujian komponen dilakukan secara terpisah antara sensor level, motor valve, sensor flow.

4.1.2.1 Pengujian Motor Valve

Pengujian motor valve dilakukan untuk melihat daya yang dibutuhkan oleh valve dan waktu untuk bekerja dari posisi tertutup menjadi terbuka atau sebaliknya. Dibawah ini adalah gambar rangkaian motor valve yang digunakan pada sistem ini.

Pengujian motor valve ini dilakukan dengan menyambungkan *driver motor* dengan *power supply*. Untuk melihat daya yang dibutuhkan oleh motor untuk menggerakkan valve. Berikut adalah data pengujian motor stepper terhadap bukaan valve.

Tabel 4. 1 Hasil Pengujian Stepper

No	Flow (LPM)	valve buka (%)
1	11	100
2	10	90
3	9	80
4	8	70
5	7	60
6	6	50
7	5	40
8	4	30

No	Flow (LPM)	valve buka (%)
9	2	10
10	0	0

4.1.2.2 Pengujian Sensor Level

Sensor Level yang digunakan, harus mampu membaca level yang dihasilkan dengan sesuai, untuk itu dilihat dari karakteristik dari sensor level ini. Pengujian sensor level ini dilakukan dengan menghubungkan sensor level dengan mikrokontroler dan diberikan sumber tegangan yang akan dibaca oleh sensor dan ditampilkan melalui LCD. Hasil pembacaan oleh sensor level dibandingkan dengan hasil pembacaan pada alat ukur berupa penggaris. Berikut ini adalah perbandingan hasil pengukuran level oleh sensor dan penggaris. Pengukuran dilakukan dengan diberikan 5 range pengukuran (5, 10, 15, 20, dan 25) secara naik dan dilanjutkan dengan pengambilan data secara turun.

Berikut ini adalah perbandingan hasil pengukuran level oleh sensor dan penggaris.

Tabel 4. 2 Pengambilan Data Naik

No	Pembacaan standar	Pembacaan Alat					Rata-Rata
		Data Ke					
		1	2	3	4	5	
1	5	5	5	6	5	4	5
2	10	9	10	10	9	11	9,8
3	15	15	15	14	15	16	15
4	20	20	19	20	20	20	19,8
5	25	24	25	25	24	25	24,6

Tabel 4. 3 Pengambilan Data Turun

No	Pembacaan standar	Pembacaan Alat					Rata-Rata
		Data Ke					
		1	2	3	4	5	
1	25	24	25	25	25	25	24,8
2	20	20	20	20	19	20	19,8
3	15	15	15	15	16	15	15,2
4	10	10	10	9	9	10	9,6
5	5	5	5	5	4	5	4,8

Tabel 4. 4 Data Kalibrasi Sensor

No	Pembacaan standar(X)	Pembacaan Alat		Rata-rata	Koreksi (Y)
		Rata-rata Naik	Rata-rata Turun		
1	5	5	4,8	4,9	-0,1
2	10	9,8	9,6	9,7	-0,3
3	15	15	15,2	15,1	0,1
4	20	19,8	19,8	19,8	-0,2
5	25	24,6	24,8	24,7	-0,3
Jumlah					-0,8
Rata-Rata					-0,16

Tabel 4. 5 Data Kalibrasi Lanjutan

No	X	X ²	Y-Y'	X*Y	(Y-Y') ²
1	5	25	0,625	-0,5	1,25
2	10	100	1,875	-3	3,75
3	15	225	-0,625	1,5	-1,25
4	20	400	1,25	-4	2,5
5	25	625	1,875	-7,5	3,75
Jumlah	75	1375	5	13,5	10
Rata-Rata	12	275	1	-2,7	5

Berikut ini merupakan cara untuk mencari nilai ketidakpastian sensor Level.

a. Nilai Ketidakpastian Tipe A

$$\sigma = \frac{\sqrt{\sum(y-\bar{y})^2}}{n-1} \dots\dots\dots (4.1)$$

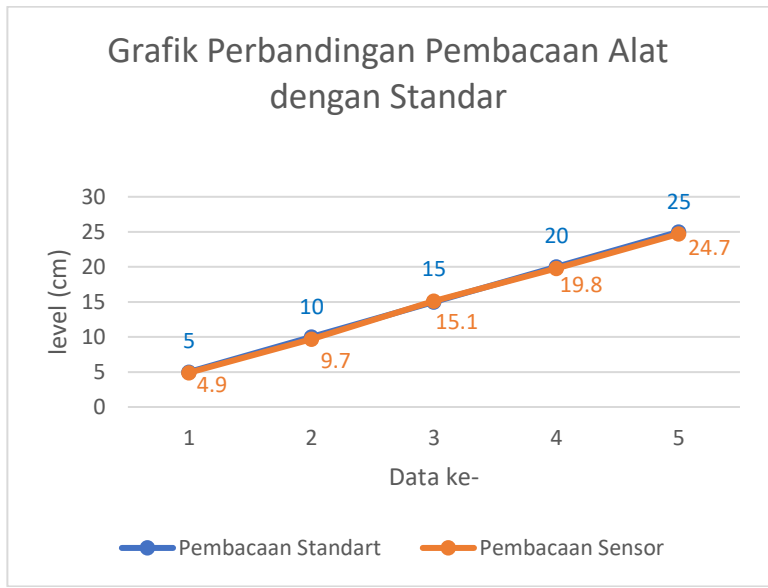
$$\sigma = \frac{\sqrt{0,16}}{5 - 1} = 0,04$$

Sehingga didapatkan nilai U_{a1} sebagai berikut:

$$U_{a1} = \frac{\sigma}{\sqrt{n}} \dots\dots\dots (4.2)$$

$$U_{a1} = \frac{0,04}{\sqrt{5}} = 0,01782e$$

Berikut ini adalah grafik perbandingan antara pembacaan oleh sensor dan penggaris.



Gambar 4. 3 Grafik Pembacaan oleh Sensor dan Penggaris

4.1.2.3 Pengujian Sensor Flow

Pengujian sensor flow membandingkan laju aliran yang dibaca oleh sensor flow dengan volume pada tangki ukur per satuan waktu. Pengambilan data dilakukan pada 5 titik dengan rentang yang sama,

Tabel 4.6 Data Pengukuran pada Tangki Ukur

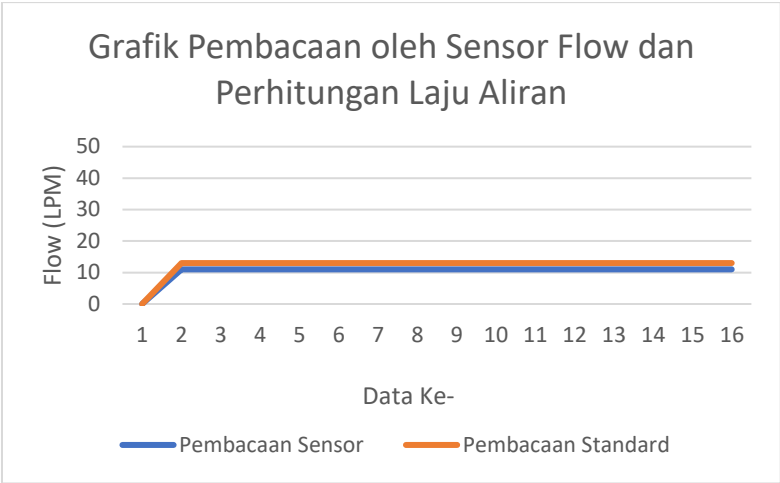
No.	Alas Tangki (cm ²)	Tinggi cairan(cm)	Volume (L)	Debit (LPM)
1	20 x 20 = 400	0	0	0
2		1	0.4	13.3
3		2	0.8	13.3
4		3	1.2	13.3
5		4	1.6	13.3
6		5	2	13.3
7		6	2.4	13.3
8		7	2.8	13.3
9		8	3.2	13.3
10		9	3.6	13.3
11		10	4	13.3
12		11	4.4	13.3
13		12	4.8	13.3
14		13	5.2	13.3
15		14	5.6	13.3
16		15	6	13.3

Berdasarkan data hasil pengukuran pada tabel diatas, diperoleh perbandingan laju aliran yang dibaca oleh sensor dengan volume pada tangki ukur per satuan waktu. Berikut merupakan tabel pembacaan sensor dan perhitungan laju aliran.

Tabel 4.7 Data Hasil Pembacaan Sensor dan Pembacaan Standard

No.	Pembacaan Sensor	Debit (LPM)
1	0	0
2	11	13
3	11	13
4	11	13
5	11	13
6	11	13
7	11	13
8	11	13
9	11	13
10	11	13
11	11	13
12	11	13
13	11	13
14	11	13
15	11	13
16	11	13

Dari tabel tersebut, diperoleh grafik perbandingan dari pembacaan sensor dan pembacaan standard. Adapun grafik tersebut seperti berikut ini :



Gambar 4.4 Grafik Pembacaan Sensor Flow dan Pembacaan Standard

4.2 Pengujian Sistem Pengendalian Level

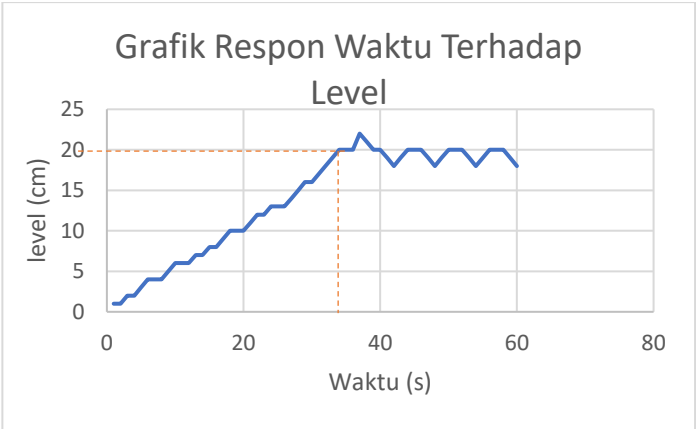
Berikut merupakan hasil pengujian sistem pengendalian level yang diambil selama 60 detik pertama. Berikut adalah data berupa tabel yang terdiri dari pembacaan level dan waktu yang dibutuhkan.

Tabel 4.8 Data Hasil Pengujian Sistem Pengendalian Level

Waktu (s)	Level (cm)	Waktu (s)	Level (cm)
1	1	31	17
2	1	32	18
3	2	33	19
4	2	34	20
5	3	35	20
6	4	36	20
7	4	37	22
8	4	38	21
9	5	39	20

10	6	40	20
11	6	41	19
12	6	42	18
13	7	43	19
14	7	44	20
15	8	45	20
16	8	46	20
17	9	47	19
18	10	48	18
19	10	49	19
20	10	50	20
21	11	51	20
22	12	52	20
23	12	53	19
24	13	54	18
25	13	55	19
26	13	56	20
27	14	57	20
28	15	58	20
29	16	59	19
30	16	60	18

Dari tabel tersebut dengan *set point* level yaitu 20 cm , diperoleh grafik hubungan respon waktu terhadap level. Adapun grafik tersebut seperti berikut ini :



Gambar 4.5 Grafik Respon Dinamik

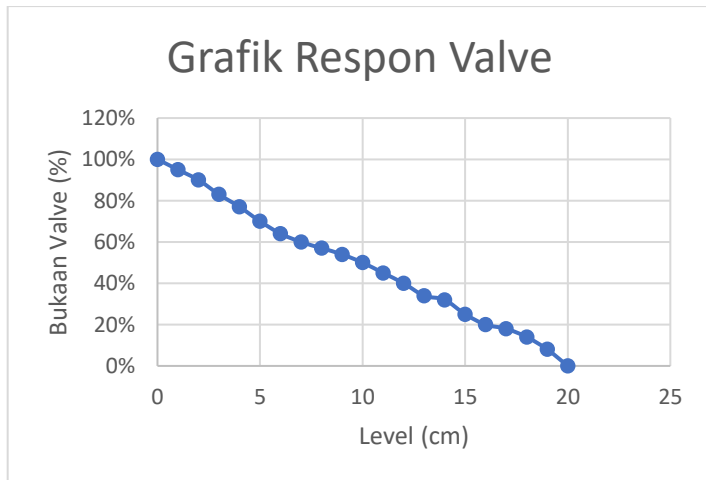
Dari data tersebut diperoleh karakteristik respon T_s (*settling time*) yaitu sebesar 34 detik. Dari data tersebut, didapatkan pula respon motor valve terhadap level. Data ini diambil pada 10 detik terakhir dari yang terekam pada saat *running* sistem. Adapun respon *valve* tersebut dibuat dalam tabel berikut ini:

Tabel 4.9 Respon *Valve* Terhadap Nilai Level

Level (cm)	Valve (%)
0	100%
1	95%
2	90%
3	83%
4	77%
5	70%
6	64%
7	60%
8	57%
9	54%
10	50%

11	45%
12	40%
13	34%
14	32%
15	25%
16	20%
17	18%
18	14%
19	8%
20	0%

Dari data yang ada pada tabel di atas, diperoleh grafik hubungan respon valve terhadap level. Adapun grafik tersebut seperti berikut ini :



Gambar 4.6 Grafik Respon Valve

Dari grafik hubungan antara level dan bukaan valve, dengan *set point* yang telah ditentukan yaitu level sebesar 20 cm,

dapat diketahui bahwa motor akan bergerak membuka ketika level kurang dari 20 cm dan ketika pembacaan level lebih dari 20 cm maka motor valve akan bergerak menutup hingga level mencapai *set point*..

4.3 Pembahasan

Pada tugas akhir sistem pengendalian level ini terdiri dari perancangan mekanik dan elektrik agar system berjalan sesuai dengan *set point*. Perancangan mekanik dilakukan dengan pembuatan tangki pengukuran dilengkapi dengan *tube* sebagai *outlet* menuju ke *storage*. Perancangan elektrik dilakukan dengan membuat sistem pengendalian level yang terdiri dari sensor, kontroler, dan aktuator. Pada sistem pengendalian level ini, sensor yang digunakan adalah Sensor *Ultrasonic* HCSR04, Atmega16 sebagai kontroler, dan Motor Valve sebagai *final control element*. Pada tugas akhir ini dilakukan pengujian masing-masing komponen agar komponen yang digunakan berjalan dengan baik dan proses pada sistem pengendalian level juga dapat berjalan sesuai dengan *set point* yang telah ditentukan. Pengujian komponen dimulai dari pengujian aktuator berupa motor valve, pengujian motor valve ini dilakukan dengan menyambungkan *driver motor* dengan *power supply*, untuk melihat daya yang dibutuhkan oleh motor untuk menggerakkan valve. Data hasil pengujian motor valve dapat dilihat pada tabel 4.1. Sensor yang akan digunakan dalam sistem pengendalian level ini, yaitu sensor *ultrasonic* HCSR04. Pengujian sensor ini dilakukan dengan cara menghubungkan sensor *ultrasonic* pada mikrokontroler agar terbaca oleh LCD yang kemudian hasil pembacaannya dibandingkan dengan alat ukur standar berupa penggaris, kemudian didapatkan data naik dan data turun. Data pengujian sensor level dapat dilihat pada tabel 4.3. Pengujian sensor flow dilakukan dengan cara membandingkan laju aliran yang dibaca oleh sensor flow dengan perhitungan volume pada tangki ukur per satuan waktu, data pengujian sensor flow dapat dilihat pada tabel 4.6. Pada tugas akhir ini didapat dua data mengenai pembacaan *sensor ultrasonic* dan respon dari Motor Valve. Dari data

pengujian alat pada sistem pengendalian level, didapatkan *respon time* yang cukup baik untuk mencapai *steady state* dengan maksimum *over shoot* 22 cm. Pengendalian level yang dilakukan yaitu mengatur input agar output sensor sesuai dengan *set point* yang ditentukan. *Set point* yang diinginkan yaitu 20 cm. Metode pengendalian yang dilakukan yaitu saat pembacaan sensor level kurang dari 20 cm, maka motor valve akan bergerak membuka dan ketika pembacaan level lebih dari 20 cm maka motor valve akan bergerak menutup hingga level mencapai *set point* yang ditentukan yaitu 20 cm. *Settling Time* atau waktu yang diperlukan untuk mencapai *set point* level yaitu kurang lebih 34 detik.

(Halaman Ini Sengaja Dikosongkan)

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan rancang bangun dan analisa data yang telah dilakukan maka dapat disimpulkan bahwa :

1. Untuk mencapai *steady state* waktu yang dibutuhkan adalah 34 detik.
2. Ketika output melebihi *set point* maka MOV akan bergerak menutup, dan ketika belum mencapai set point maka MOV akan bergerak membuka.

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan adalah memastikan bahwa plant sudah tidak ada yang bocor untuk menghindari *losse* saat pengendalian karena akan mempengaruhi laju aliran dan membuat pengukuran level menjadi tidak stabil.

DAFTAR PUSTAKA

- [1] Bakti, Dheka, dkk. (2013). *PENGENDALIAN LEVE MENGGUNAKAN PENGENDALI PROPORSIONAL BERBASIS ATMEGA*. Surabaya: ITS.
- [2] WingWijayanto. (2012). *RANCANG BANGUN SISTEM PENGENDALIAN LEVEL DAN FLOW*.
- [3] Punchstein, A.F., Liroyd,T.C., Conrad, A.G., 1960. "Alternating Current Machines". New York: John Willey and Sons Inc.
- [4] Anonim. 2016. *Modul Praktikum Sistem Pengukuran Aliran*. Surabaya : Departemen Teknik Instrumentasi, FV-ITS
- [5] Nurcahyo. 2012. Aplikasi dan Teknik Pemrograman Mikrokontroler AVR Atmel CV Andi Offset.
- [6] Dr. Sri Purnomo Sari, S. M. (n.d.). PENGARUH JARAK DAN UKURAN NOZZLE.
- [7] Texas Instrument. 2014.Hall-Effect Sensor with PNP or NPN Output Manual Book.
- [8] Politeknik Negeri Elektronika Surabaya.
bima.lecturer.pens.ac.id/materi%20praktikum%20dsp/ps2_c odec_sampling.pdf. n.d. bima.lecturer.pens.ac.id (accessed January 29, 2018).

LAMPIRAN

[1] Ultrasonic Ranging Module HC - SR04

☐ Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact

measurement function, the ranging accuracy can reach to 3mm. The modules

includes ultrasonic transmitters, receiver and control circuit. The basic principle

of work:

(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz and detect whether there is a

pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is

the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

☐ Wire connecting direct as following:

5V Supply

Trigger Pulse Input

Echo Pulse Output

0V Ground

Electric Parameter

Working Voltage DC 5 V

Working Current 15mA

Working Frequency 40Hz

Max Range 4m

Min Range 2cm

MeasuringAngle 15 degree

Trigger Input Signal 10uS TTL pulse

Echo Output Signal Input TTL lever signal and the range in

proportion

Dimension 45*20*15mm

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS

pulse to the trigger input to start the ranging, and then the module will send out

an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a

distance object that is pulse width and the range in proportion .You can

calculate the range through the time interval between sending trigger signal and

receiving echo signal. Formula: $\text{uS} / 58 = \text{centimeters}$ or $\text{uS} / 148 = \text{inch}$; or: the

range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms

measurement cycle, in order to prevent trigger signal to the echo signal.

□ Attention:

The module is not suggested to connect directly to electric, if connected

electric, the GND terminal should be connected the module first, otherwise,

it will affect the normal work of the module.

When tested objects, the range of area is not less than 0.5 square meters

and the plane requests as smooth as possible, otherwise ,it will affect the

results of measuring.

[2] *Datasheet Water Flow Sensor*

Introduction Water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls. Its speed changes with different rate of flow.

The hall-effect sensor outputs the corresponding pulse Signal.
Model:POW110D3B

Specification Working voltage 5V-24V, Maximum current 15 mA (DC 5V), Weight 43 g External diameters 20mm Flow rate range 1~30 L/min Operating temperature 0°C~80°C Liquid temperature.

[3]

```

/*****
*

```

This program was produced by the

CodeWizardAVR V2.05.0 Professional

Automatic Program Generator

© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 7/17/2018

Author : NeVaDa

Company :

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 16.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

/

```
#include <mega16.h>
```

```
// Alphanumeric LCD Module functions
```

```
#include <alcd.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
#define stepB PORTA.6
```

```
#define stepD PORTA.4
```

```
#define stepC PORTA.5
```

```
#define stepA PORTA.7
```

```
#define Valve_safety PORTA.0
```

```
#define Valve_out PORTA.1
```

```
#define pump      PORTA.2
#define pump2     PORTA.3

#define ledrun    PORTD.7
#define buzzer    PORTD.6
#define ultrasound PORTC.1
#define maxstep   1000
```

```
int pulse,minperiode,periode,respon,direct,stepcount,level;
bit refresh,stepper;
unsigned int range;
```

```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
if (PIND.2==1)
{
    TCCR2=0x02;
    TCNT2=0x00;
    range=0;
```

```

    }
    else
    {
        TCCR2=0x00;
        TCNT2=0x00;
        level=48-((range+TCNT2)/116);
        if (level<=0)level=0;
    }
}

```

```

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    // Place your code here

    periode=pulse;
    minperiode=TCNT0;
    TCNT0=0;
    pulse=0;
}

```

```

#ifndef RXB8

```

```
#define RXB8 1
```

```
#endif
```

```
#ifndef TXB8
```

```
#define TXB8 0
```

```
#endif
```

```
#ifndef UPE
```

```
#define UPE 2
```

```
#endif
```

```
#ifndef DOR
```

```
#define DOR 3
```

```
#endif
```

```
#ifndef FE
```

```
#define FE 4
```

```
#endif
```

```
#ifndef UDRE
```

```
#define UDRE 5
```

```
#endif
```

```
#ifndef RXC
```

```
#define RXC 7
```

```
#endif
```

```
#define FRAMING_ERROR (1<<FE)
```

```
#define PARITY_ERROR (1<<UPE)
```

```
#define DATA_OVERRUN (1<<DOR)
```

```
#define DATA_REGISTER_EMPTY (1<<UDRE)
```

```
#define RX_COMPLETE (1<<RXC)
```

```
// USART Receiver buffer
```

```
#define RX_BUFFER_SIZE 4
```

```
char rx_buffer[RX_BUFFER_SIZE];
```

```
#if RX_BUFFER_SIZE <= 256
```

```
unsigned char rx_wr_index,rx_rd_index,rx_counter;
```

```
#else
```

```
unsigned int rx_wr_index,rx_rd_index,rx_counter;
```

```
#endif
```

```
// This flag is set on USART Receiver buffer overflow
```

```
bit rx_buffer_overflow;
```

```

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
    char status,data;
    status=UCSRA;
    data=UDR;
    if(((data=='#')||(data=='*'))          // Inisial stx (0x23 = #)
    {
        rx_counter=0;                    // rx_counter mulai dari 0
        rx_wr_index=0;                   // rx_wr_index mulai dari 0
    }
    if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
    {
        rx_buffer[rx_wr_index++]=data;
#ifdef RX_BUFFER_SIZE == 256
        // special case for receiver buffer size=256
        if (++rx_counter == 0)
        {
            #else
            if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
            if (++rx_counter == RX_BUFFER_SIZE)
            {

```

```

        rx_counter=0;
#endif

        rx_buffer_overflow=1;
    }
}

}

#ifdef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    #if RX_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;

```



```
}
```

```
#pragma used-
```

```
#endif
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

```
// Timer 0 overflow interrupt service routine
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
pulse++;
```

```
}
```

```
// Timer1 overflow interrupt service routine
```

```
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
respon++;
```

```
if (respon>=122)
```

```
{
```

```
refresh=1;
```

```
respon=0;
```

```

    }

    stepper=1;
}

// Timer2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
    // Place your code here
    range=range+256;
    if (range>=9280)
    {
        TCCR2=0x00;
        level=48;}
}

moving_stepper(int i)
{
    if (direct==1) //valve buka
    { switch (i)
    {
        case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
        case 1: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 2: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
    }
    }
}

```

```

        case 3: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 4: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 5: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 6: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 7: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 8: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
    }
    stepcount++;if
(stepcount>=maxstep){direct=0;stepcount=maxstep;stepA=0;
stepB=0; stepC=0; stepD=0;}
}
else if(direct==2) //valve tutup
{
switch (i)
{

        case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
        case 1: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 2: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 3: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 4: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 5: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 6: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 7: {stepA=0; stepB=0; stepC=0; stepD=1; break;}

```

```

        case 8: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
    }
    ;if(stepcount<=0){stepcount=0;direct=0;stepA=0;          stepcount--
    stepC=0; stepD=0;}          stepB=0;

    }

    else if (direct==0){stepA=0; stepB=0; stepC=0; stepD=0;}
//valve mati

}

// Declare your global variables here


void main(void)

{

// Declare your local variables here

    int  valve_open,frek,p,flow,setpoint;

    unsigned char temp[16], skip[3];

// Input/Output Ports initialization

// Port A initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0

PORTA=0x0F;

DDRA=0xFF;

```

```
// Port B initialization
```

```
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out  
Func2=Out Func1=Out Func0=Out
```

```
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0  
State1=0 State0=0
```

```
PORTB=0x00;
```

```
DDRB=0xFF;
```

```
// Port C initialization
```

```
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out  
Func2=Out Func1=Out Func0=Out
```

```
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0  
State1=0 State0=0
```

```
PORTC=0x00;
```

```
DDRC=0xFF;
```

```
// Port D initialization
```

```
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In  
Func2=In Func1=In Func0=In
```

```
// State7=0 State6=0 State5=0 State4=0 State3=P State2=T  
State1=T State0=T
```

```
PORTD=0x08;
```

```
DDRD=0xF0;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 62.500 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x02;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 250.000 kHz
// Mode: Fast PWM top=OCR1A
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x03;
```

```
TCCR1B=0x1B;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x04;
```

```
OCR1AL=0xE2;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 2000.000 kHz
```

```
// Mode: Normal top=0xFF
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x02;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: On
// INT0 Mode: Any change
// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
GICR|=0xC0;
MCUCR=0x09;
MCUCSR=0x00;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x45;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
```



```
UBRRL=0x67;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// ADC initialization
```

```
// ADC disabled
```

```
ADCSRA=0x00;
```

```
// SPI initialization
```

```
// SPI disabled
```

```
SPCR=0x00;
```

```
// TWI initialization
```

```
// TWI disabled
```

```
TWCR=0x00;
```

```
// Alphanumeric LCD initialization
```

```
// Connections specified in the
```

```
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD  
menu:
```

```
// RS - PORTB Bit 0
```

```
// RD - PORTB Bit 1
```

```
// EN - PORTB Bit 2
```

```
// D4 - PORTA Bit 4
```

```
// D5 - PORTA Bit 5
```

```
// D6 - PORTA Bit 6
```

```
// D7 - PORTA Bit 7
```

```
// Characters/line: 20
```

```
lcd_init(20);
```

```
// Global enable interrupts
```

```
#asm("sei")
```

```
lcd_clear();
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("Simulator Level");
```

```
lcd_gotoxy(0,1);
```

```
lcd_putsf("Naimah & Nafik");
```

```
level=0;
```

```
stepcount=maxstep;direct=2;//tutup valve
```

```
p=0;
```

```

while(stepcount>0)
{ if (stepper==1)
    { p++;
      if (p>=9) p=1;
      moving_stepper(p);stepper=0;
    }
};
ledrun=1;
direct=0;
lcd_clear();
setpoint=0;
pump2=0;
pump=1;
flow=0;
while (1)
    { if(rx_buffer_overflow==1) //penanda
      flag awal data
      {
          if
          ((rx_buffer[0]=='*')&&(rx_buffer[1]=='0')&&(rx_buffer[2]=='0')
          &&(rx_buffer[3]=='0'))pump=1;

          else if
          ((rx_buffer[0]=='#')&&(rx_buffer[1]=='1')&&(rx_buffer[2]=='1')
          &&(rx_buffer[3]=='1'))pump=0;

          else if (rx_buffer[0]=='@')

```

```

    {
        skip[0]=rx_buffer[1];
        skip[1]=rx_buffer[2];
        skip[2]=rx_buffer[3];
        setpoint=atoi(skip);

    }
    rx_buffer_overflow=0;
}
if (stepper==1)
{
    p++;
    if (p>=9)
    {
        p=1;

        // printf("%c%4d%c", '$',flow,'#');
    }

    moving_stepper(p);
    stepper=0;

}

```

// Place your code here

```

if (refresh==1)
{

//printf("%c%4d%c",'@',level,'#');
printf ("%d;", level);
lcd_gotoxy(0,0);
lcd_putsf("Level = ");
sprintf(temp,"%2d cm    ",level);
lcd_puts(temp);


if((periode==0)&&(minperiode==0))frek=0; //flow
frek=1/((periode*0.000128)+(minperiode*0.0000005));


flow=frek*0.13;//dikali dengan hasil kalibrasi


lcd_gotoxy(0,1);
lcd_putsf("Flow = ");//tampilan lcd
sprintf(temp,"%4d flow",flow);
lcd_puts(temp);
printf("%d", flow);


valve_open=stepcount/(maxstep/100);
lcd_gotoxy(0,2);

```

```

lcd_putsf("Valve = ");
sprintf(temp,"%3d %c",valve_open,'%');
printf("%d;\r\n",valve_open);
lcd_puts(temp);

```

```

ultrasound=0;
ultrasound=1;
delay_us(10);
ultrasound=0;

```

```

ledrun=1;

//progam penegndalian motor stepper
if (setpoint>=valve_open+2)direct=1;
else if (setpoint<=valve_open-2)direct=2;
else if
((setpoint<valve_open+2)&&(setpoint>valve_open-2))direct=0;

```

```

//progam setpoint valve
if (level>=20+1){ setpoint=0;}
else if (level<=20-1)setpoint=100;
if (level>=20+5)Valve_safety=1;
else Valve_safety=0;

```

```
if (level>=10)Valve_out=1;
```

```
refresh=0;
```

```
}
```

```
}
```

```
}
```

y 29, 2018).

BIODATA



Zumrotun Naimah yang biasa dipanggil “Naimah” ini merupakan mahasiswa dari daerah Lamongan. Penulis menempuh pendidikan formal di SDN Kedungrejo 1, SMP 1 Babat Lamongan, dan SMAN 1 GIRI Banyuwangi. Pada tahun 2015 penulis mengikuti test ujian D3 ITS dan diterima pada jurusan D3 Metrologi dan Instrumentasi ini. Apabila ada pertanyaan mengenai tugas akhir penulis dapat menghubungi 083832028206 atau melalui email : nsoecipto@gmail.com